Administration Guide Supplemental

SSL/TLS Certificate Check for GEMS and Good Work

Product Version: 2.0

Issued: 30-Nov-15 | Last Updated: 30-Nov-15

Good Enterprise Mobility Server™

Legal Notice

This document, as well as all accompanying documents for this product, is published by Good Technology Corporation ("Good"). Good may have patents or pending patent applications, trademarks, copyrights, and other intellectual property rights covering the subject matter in these documents. The furnishing of this, or any other document, does not in any way imply any license to these or other intellectual properties, except as expressly provided in written license agreements with Good. This document is for the use of licensed or authorized users only. No part of this document may be used, sold, reproduced, stored in a database or retrieval system or transmitted in any form or by any means, electronic or physical, for any purpose, other than the purchaser's authorized use without the express written permission of Good. Any unauthorized copying, distribution or disclosure of information is a violation of copyright laws.

While every effort has been made to ensure technical accuracy, information in this document is subject to change without notice and does not represent a commitment on the part of Good. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those written agreements.

The documentation provided is subject to change at Good's sole discretion without notice. It is your responsibility to utilize the most current documentation available. Good assumes no duty to update you, and therefore Good recommends that you check frequently for new versions. This documentation is provided "as is" and Good assumes no liability for the accuracy or completeness of the content. The content of this document may contain information regarding Good's future plans, including roadmaps and feature sets not yet available. It is stressed that this information is non-binding and Good creates no contractual obligation to deliver the features and functionality described herein, and expressly disclaims all theories of contract, detrimental reliance and/or promissory estoppel or similar theories.

Legal Information

© Copyright 2015. All rights reserved. All use is subject to license terms posted at www.good.com/legal. GOOD, GOOD TECHNOLOGY, the GOOD logo, GOOD FOR ENTERPRISE, GOOD FOR GOVERNMENT, GOOD FOR YOU, GOOD APPCENTRAL, GOOD DYNAMICS, SECURED BY GOOD, GOOD MOBILE MANAGER, GOOD CONNECT, GOOD SHARE, GOOD TRUST, GOOD VAULT, and GOOD DYNAMICS APPKINETICS are trademarks of Good Technology Corporation and its related entities. All third-party technology products are protected by issued and pending U.S. and foreign patents.

Patent Information: https://www1.good.com/legal/other-legal.html#trademark

Revision History

Log begins 31-Aug-15

Date	Description
31-Aug-15	GW 1.5 MR edition published
23-Sep-15	Added clarification under "Java Keystore" stating that, while the Presence service uses JKS for communications with GD, it uses the Windows keystore for communicating with Lync
30-Sep-15	Corrected exported certificate output file extension under "Exporting the GEMS Self-Signed Certificate" to **.cer** (was incorrectly cited as **.crt**)

Table of Contents

Abstract 1
Keystores and Certificates for GEMS Services 2
Certificate Keystores 2
Java Keystore 3
Windows Keystore 3
Certificates Used by GEMS to Authenticate Third-Party Servers 4
Active Directory (AD) 4
Exchange 4
Lync 4
SharePoint 4
Office Web App Server (OWAS) 5
Good Proxy 5
Good Network Operations Center (NOC) 5
Disabling SSL Certificate Checking in GEMS 5
Keystores and Certificates for Good Work 6
Certificate Keystores 6
GD Keystore 6
Device Keystore 6
Certificates Used by Good Work to Authenticate Third-Party Servers 7
Good Proxy 7
Good NOC 7
GEMS 7
Exchange 7
Disabling SSL Checking in Good Work 7
Importing Certificates into the Java Keystore 8
Importing Certificates into the Windows Keystore 9
Exporting the GEMS Self-Signed Certificate 10

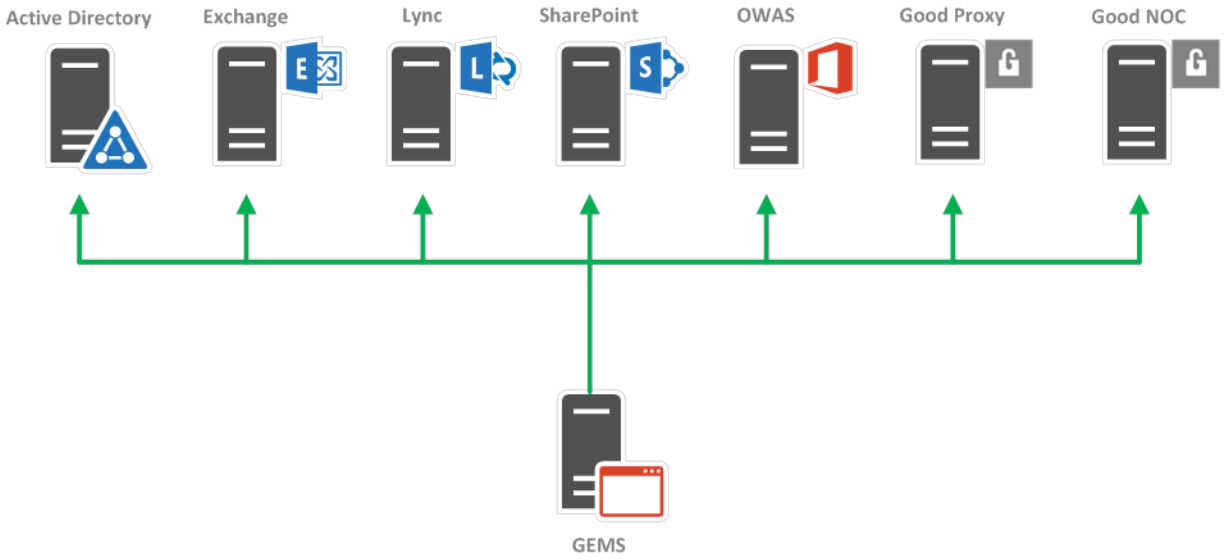
Glossary 13

Abstract

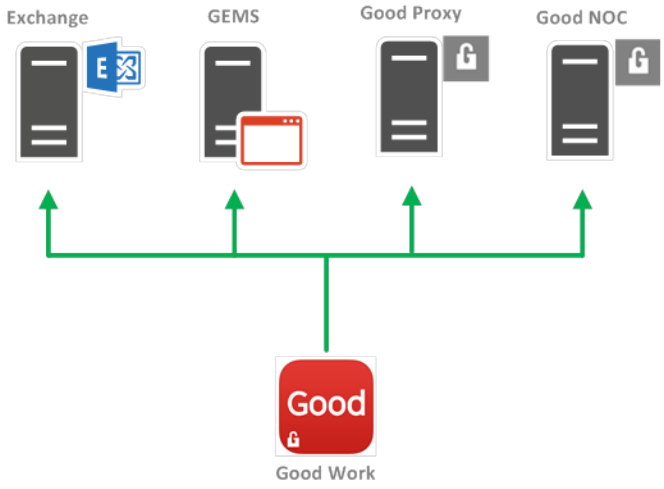
Transport Layer Security (**TLS** and its predecessor, Secure Socket Layer (**SSL**), are cryptographic protocols designed to furnish secure communications over a computer network using X.509 certificates. In cryptography, X.509 is an ITU-T standard for a public key infrastructure (PKI) and Privilege Management Infrastructure (PMI). X.509 specifies, among other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.

Good Enterprise Mobility Server (GEMS) and Good Work use various SSL certificates to authenticate with third-arty systems. For the authentication process to work, each party must trust the others SSL certificate.

The following illustrations depict the variety of systems to which GEMS and Good Work connect.



GEMS SSL/TLS Connections



Good Work SSL/TLS Connections

A **keystore** is a repository of security certificates—either authorization certificates or public key certificates—protecting each private key with its individual password, as well as protecting the integrity of the entire keystore with a password.

Keystores and Certificates for GEMS Services

A GEMS host machine provides multiple services, currently including:

1. Core (Dashboard)
2. Push Notifications
3. Presence
4. Instant Message (IM)
5. Docs
6. Directory Lookup
7. Certificate Lookup

For more information on what each service provides and how each is configured, please see the *GEMS Administration Guide for Administrators* (a k a "GEMS Administration Guide" or "GEMS Admin Guide") available from the Good Admin Portal.

Here, we will limit discussion to how GEMS services use SSL to authenticate with other servers.

Certificate Keystores

By default, when GEMS attempts an outbound SSL connection to a third-party server, it performs a check on the other server's SSL certificate before connecting.

The SSL validation process checks for two essential attributes:

(a) that the certificate has a verifiable certificate path, and

(b) that the requested fully qualified domain name (FQDN) matches the FQDN of the certificate.

Depending on the GEMS service making the request, one or both of two types of keystore—Java or Windows—is used.

Java Keystore

These GEMS services use the Java keystore (JKS):

- **Push Notifications**
- **Presence** – for communication with Good Proxy; for communication with Lync, the Windows keystore is used.
- **Docs**
- **Directory Lookup**
- **Certificate Lookup**

The JKS default location is **C:\Program Files\Java\jre7\lib\security\cacerts**.

The default password for the keystore is "**changeit**".

Note: The default path may differ depending on the version of Java you are using.

By default, the Java keystore only contains common public certificate authorities. This means that GEMS will not be able to connect to any third-party servers not using publicly verifiable certificates unless the default Java keystore is updated.

Make sure that when updating the Java keystore, you:

a. Import all relevant third-party CA certificates into the java keystore

b. Ensure that the requested FQDN by GEMS matches the FQDN in the 3rd party server's certificate

See Importing CA Certificates into the Java Keystore below for additional guidance.

Windows Keystore

These GEMS services use the Windows keystore:

- Presence
- Instant Message (Connect)

The Windows keystore can be accessed from the MMC window on the GEMS server. By default, the Windows keystore only contains common public certificate authorities. Depending on how group policies are configured on the GEMS server, however, the Windows keystore may also contain third-party certificate authorities.

Just like the Java keystore, it is best to check the Windows keystore first and then import any missing CA certificates. Always make sure the requested FQDN by GEMS matches the FQDN in the third-party server's certificate.

See Importing CA certificates into the Windows Keystore below for guidance.

Certificates Used by GEMS to Authenticate Third-Party Servers

Now let's take a look at the various servers used by GEMS and the changes needed in order for GEMS to trust third-party servers.

Active Directory (AD)

The GEMS DOCS and Certificate Lookup services require direct access to AD. If you are using SSL for LDAP, then:

- If your LDAP certificate is signed by an internal CA, you must export your CA certificate and import it into GEMS Java keystore
- Changes to the GEMS Java keystore require a restart of the **Good Technology Common** Service.

Exchange

The GEMS Notification and Directory Lookup services require direct access to Exchange. Please note the following:

- If your Exchange server is using a self-signed certificate, you must export the self-signed certificate and import it into the GEMS Java keystore
- If your Exchange server certificate is signed by an internal CA, you must export the CA certificate and import it into the GEMS Java keystore
- Any changes to the GEMS Java keystore require a restart of the **Good Technology Common** service.

Lync

The GEMS Presence and Instant Messaging services both require direct access to Lync. The GEMS Windows keystore should already have the Lync CA certificate. In most cases, updates to the GEMS Windows keystore are unnecessary. However, if the Lync CA certificate does not exist in the GEMS Windows keystore, then in order for GEMS to trust Lync, you must import it into the GEMS Windows keystore.

SharePoint

Because the GEMS Docs service requires direct access to SharePoint, please note the following:

- If your SharePoint server is using a self-signed certificate, you must export the self-signed certificate and import it into the GEMS Java keystore
- If your SharePoint server certificate is signed by an internal CA, you must export the CA certificate and import it into the GEMS Java keystore
- Any changes to the GEMS Java keystore require a restart of the **Good Technology Common** service.

Office Web App Server (OWAS)

GEMS Docs also needs direct access to your OWAS server, wherein the following conditions must be met:

- If your OWAS server is using a self-signed certificate, export the self-signed certificate and import it into the GEMS Java keystore
- If your OWAS server certificate is signed by an internal CA, export the CA certificate and import it into the GEMS Java keystore
- Any change to the GEMS Java keystore requires a restart of the **Good Technology Common** service.

Good Proxy

The Good Proxy server certificate is signed by an internal certificate authority (Good Control). If you configure GEMS to connect to Good Proxy via SSL, then you must do the following in order for GEMS to trust Good Proxy:

1. Export the Good Proxy CA certificate and import it into the GEMS Java keystore.
2. Restart the **Good Technology Common** service.

See "Importing CA Certificates for GEMS" in the GEMS Admin Guide for guidance on exporting the Good Proxy CA certificate.

Good Network Operations Center (NOC)

The Good NOC uses public certificates. GEMS will trust it by default. Therefore, no keystore updates are needed.

Disabling SSL Certificate Checking in GEMS

Disabling the automatic SSL check in GEMS should be done in a test or proof of concept (POC) environment only.

Currently, disabling SSL checking is configured via a global parameter from the GEMS Web Console at https://localhost:8443/system/console. The default login is **admin/admin**. From **OSGi** > **Configuration** > **Good Technology Async HTTP Client Configuration**, select **Disable SSL certificate checking**.

Subsequent releases of GEMS will make this parameter available for each GEMS service directly from the GEMS Dashboard.

Keystores and Certificates for Good Work

The Good Work collaboration client consists of multiple components. Major components currently comprise:

1. Email
2. Calendar
3. Contact Search
4. Document sharing

For more information on each component and how it is configured, see the GEMS Admin Guide and the *Good Work Product Guide*. The rest of this section will examine how these Good Work components uses SSL to authenticate with other servers.

Here, similar to GEMS, we will limit discussion to how Good Work components use SSL to authenticate with other servers.

Certificate Keystores

When Good Work makes an outbound SSL connection to a third-party server, it performs a SSL check on the third- party server's SSL certificate before connecting.

that the SSL validation process checks for two essential attributes:

(a) The certificate must have a verifiable certificate path

(b) that the requested FQDN must match the FQDN of the certificate

Good Work has access to two different keystores for the SSL validation process: a Good Dynamics (GD) keystore and the device keystore. Depending on the security configuration in Good Control (under Policy Sets), Good Work uses one or the other or both keystores for certificate validation. It uses both by default.

GD Keystore

The GD keystore is located in the Good Work secure container. There is no direct access to this keystore. Importing certificates to this keystore is done from Good Control (under **Certificates** > **Server Certificates**). Any server certificates uploaded to Good Control are automatically distributed to all GD apps, including Good Work.

Device Keystore

The device keystore contains common public certificate authorities. This keystore is unique to the device. Refer to your device user manual to identify which certificate authorities are included and how to modify the keystore.

Certificates Used by Good Work to Authenticate Third-Party Servers

Now let's take a look at the keystore changes needed to establish trust between Good Work and third-party/other Good servers.

Good Proxy

Although Good Proxy uses an internally signed certificate, Good Work will trust Good Proxy by default because it is aware of the certificate authority used by Good Proxy. Consequently, no keystore updates are necessary.

Good NOC

Because the Good NOC uses publicly verifiable certificates, Good Work will trust the Good NOC. No keystore update is needed.

GEMS

GEMS is the Good Work proxy to critical network services (Presence, Notifications, etc.), so it is vital that Good Work trust GEMS. To ensure this trust, you must do one of the following:

a. Replace the GEMS default self-signed certificate with a publicly verifiable certificate, or
b. Export the GEMS self-signed certificate and upload it to Good Control.

For guidance on replacing the default self-signed certificate, see "Replacing the Auto-Generated Self-Signed SSL Certificate" in the GEMS Admin Guide.

See Exporting the GEMS Self-Signed Certificate below for guidance on exporting the GEMS self-signed certificate.

Exchange

Good Work connects to Exchange in order to synchronize email, calendar, contact, etc. If your Exchange server is not using a publicly verifiable certificate, you must do one of the following:

a. If your Exchange server is using a self-signed certificate, export your Exchange certificate and upload it to Good Control
b. If your Exchange server is using a certificate signed by an internal CA, export your CA certificate and upload it to Good Control

In addition to the above, you must also make sure the Exchange FQDN configured for Good Work matches the FQDN of your Exchange certificate. If the FQDNs do not match, Good Work will not trust Exchange.

Disabling SSL Checking in Good Work

Disabling SSL checking in Good Work should be done in a test or proof of concept (POC) environment only. The setting is in Good Control and determined by the value (**`true`** or **`false`**) of the **disableSSLCertificateChecking**

JSON parameter.

For more information on how to configure this setting, see "Adding the JSON Configuration for EAS" in the *Good Work Product Guide*.

Importing Certificates into the Java Keystore

Included with Java, Java keytool is a key and certificate management tool that is used to manipulate Java Keystores. Identified by an alias, each keystore entry consists of keys and certificates that form a trust chain.

To import SSL certificates into a server's JKS using Java keytool, take the following steps:

1. Locate the Java keystore.

   By default, the JKS used by GEMS is located in **C:\Program Files\Java\jre7\lib\security\cacerts**.

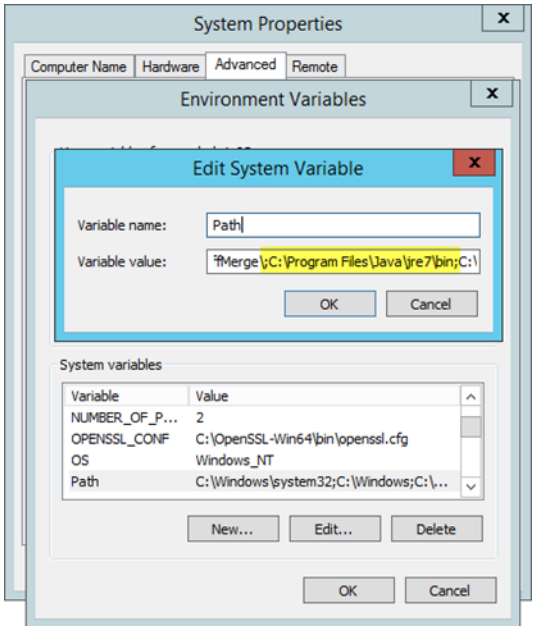
   The default path may differ depending on the version of Java you're using. Check the **JAVA_HOME** environment variable on the GEMS host to determine the location if it is not found in default directory. **JAVA_HOME** also shows which Java version GEMS is using.

   The default password for the JKS is **changeit**. Make sure to back up the keystore before making any changes to it. To back up the keystore, simply make a copy of the file.

2. Locate the Java keytool.

   The default location is **C:\Program Files\Java\jre7\bin\keytool.exe**.

3. Add the **keytool.exe** path to your **Path** environment variable.
   a. Select **Computer** from the **Start** menu.
   b. Choose **System Properties**, then click **Advanced system settings**.
   c. Open the **Advanced** tab and click **Environment Variables...**
   d. Double-click **Path** to edit it, then append the current Path variable with a semicolon followed by the path to **keytool.exe**.
   e. Click **OK**.



4. Obtain a copy of the certificate(s) you want GEMS to trust. Consult your system administrator for assistance.
5. Copy the certificates ytou want to import over to a convenient location on the GEMS host (e.g., **C:\certs**).
6. Import the certificate by taking the following steps from the GEMS host:
   a. Open a CMD prompt and change directory to the Java keystore location.
   b. Run the following command:

      ```
      keytool -import -trustcacerts -alias <cert_alias> -file c:\certs\<cert_file_name>.cer -keystore cacerts
      ```

      The *cert_alias* is arbitrary, but *cert_filename* must be the full path the certificate you want to import.
   c. Verify that the certificate was successfully imported using the following command:

      ```
      keytool -list -v -alias <cert_alias> -keystore cacerts
      ```

   d. For each certificate you want to import, repeat Steps (b) – (c).
7. Restart the **Good Technology Common** service.

Importing Certificates into the Windows Keystore

As a rule, you should only import certificates obtained from trusted sources. Importing an unreliable certificate could compromise the security of any system component that uses the imported certificate.

You can import a certificate into any logical or physical store. In most cases, you will import certificates into the **Personal** store or the **Trusted Root Certification Authorities** store, depending on whether the certificate is intended for you or if it is a root certification authority (CA) certificate.

Users or local **Administrators** is the minimum group membership required to complete this procedure.

To import a certificate:

1. In MMC, open the **Certificates snap-in** for a user, computer, or service.

   Note: If the snap-in is not already installed, see Add the Certificates Snap-in to an MMC.

2. In the console tree, click the logical store where you want to import the certificate.
3. On the **Action** menu, point to **All Tasks**, then click **Import** to start the **Certificate Import Wizard**.
4. Type the file name containing the certificate to be imported, or click **Browse** and navigate to the file.
5. If it is a PKCS #12 file:
   a. Type the password used to encrypt the private key.
   b. To be able to back up or transport your keys at a later time, enable **Mark key as exportable**.
6. Place the certificate in the appropriate store using one of the following methods:
   a. If you want the certificate automatically placed in a certificate store based on the type of certificate, click **Automatically select the certificate store based on the type of certificate**.
   b. If you want to specify where the certificate is stored, select **Place all certificates in the following store**, then click **Browse**, and choose a certificate store.

Bear in mind that the file from which you import certificates will remain intact after you have completed importing the certificates. You will be wise to delete the file if it is no longer needed.

Exporting the GEMS Self-Signed Certificate

To export the GEMS self-signed SSL certificate to another server's JKS using Java keytool, take the following steps:

1. Locate the GEMS Java keystore.

   The default location is **C:\Program Files\Good Technology\Good Enterprise Mobility Server\Good Server Distribution\gems-quickstart-*<version>*\etc\keystores\gems.jks**.

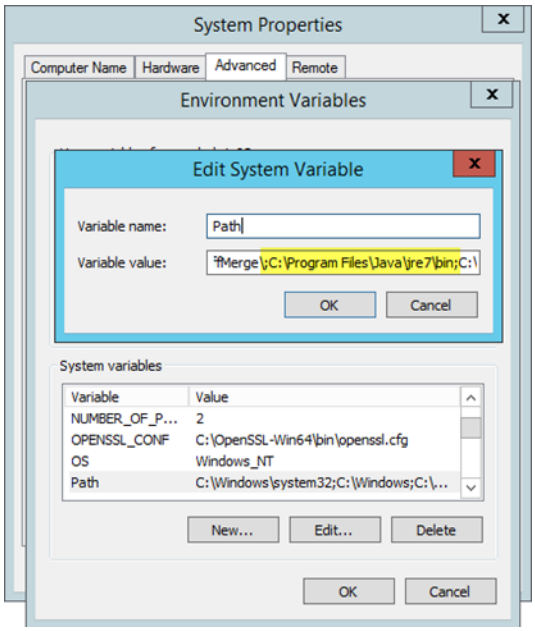
   The default path may differ depending on the GEMS version you're using.

   The default password for **gems.jks** is **changeit**. Be sure to back up the keystore before making changes. To back up the keystore, simply make a copy of the file.

2. Locate the Java keytool.

   The default location is **C:\Program Files\Java\jre7\bin\keytool.exe**.

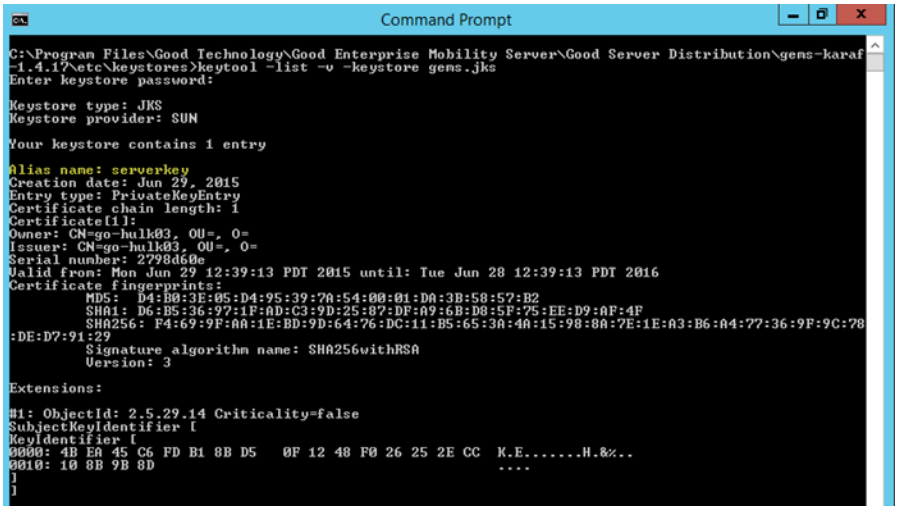
3. Add the **keytool.exe** path to your **Path** environment variable.
   a. Select **Computer** from the **Start** menu.
   b. Choose **System Properties**, then click **Advanced system settings**.
   c. Open the **Advanced** tab and click **Environment Variables...**
   d. Double-click **Path** to edit it, then append the current Path variable with a semicolon followed by the path to **keytool.exe**.
   e. Click **OK**.



4. Export the certificate by taking the following steps from the GEMS host:
   a. Open a CMD prompt and change directory to the Java keystore location.
   b. Run the following command to list the certificates in the keystore:

```
keytool -list -v -keystore gems.jks
```

   This will produce:


Command Prompt
C:\Program Files\Good Technology\Good Enterprise Mobility Server\Good Server Distribution\gems-karaf-1.4.17\etc\keystores>keytool -list -v -keystore gems.jks
Enter keystore password:
Keystore type: JKS
Keystore provider: SUN
Your keystore contains 1 entry
Alias name: serverkey
Creation date: Jun 29, 2015
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=go-hulk03, OU=, O=
Issuer: CN=go-hulk03, OU=, O=
Serial number: 2798d60e
Valid from: Mon Jun 29 12:39:13 PDT 2015 until: Tue Jun 28 12:39:13 PDT 2016
Certificate fingerprints:
MD5: D4:B0:3E:05:D4:95:39:7A:54:00:01:DA:3B:58:57:B2
SHA1: D6:B5:36:97:1F:AD:C3:9D:25:87:DF:A9:6B:D8:5F:75:EE:D9:AF:4F
SHA256: F4:69:9F:AA:1E:BD:9D:64:76:DC:11:B5:65:3A:4A:15:98:8A:7E:1E:A3:B6:A4:77:36:9F:9C:78:DE:D7:91:29
Signature algorithm name: SHA256withRSA
Version: 3
Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 4B EA 45 C6 FD B1 8B D5 0F 12 48 F0 26 25 2E CC K.E.....H.&%..
0010: 10 8B 9B 8D ....
]
]

Note the **Alias name**. Initially, and unless you change it, this value is **serverkey**

c. Export the certificate from the keystore using the following command:

```
keytool -export -alias serverkey -file gems.cer -keystore gems.jks
```

The output file is **gems.cer**.

You can now import the certificate to Good Control, in accordance with the conditions outlined in Certificates Used by Good Work to Authenticate Third-Party Servers.

Glossary

A

Access Key
Part of the activation key that is different for every GD application activation. Access keys consist of 15 letters and numbers. Access keys are generated by the enterprise GC server.

Activation Key
All the credentials necessary for activation of a GD application for an end user. The necessary credentials are a provisioning ID and an access key.

AD
Active Directory

ADSI
Active Directory Services Interface

ADT Plugin
Android Development Tools Plugin

Affinities
The feature that enables enterprises to allocate their GP servers between their GC servers and their application servers. Allocation can be an absolute division, or based on a priority order, or both.

Application Policies
The feature that enables GD application developers to add policies that are specific to their application to a GC server. Application policies are defined by developers, using an XML file format.

Application-Based Service
A GD shared service that is provided by GD applications. An application-based service uses Good Dynamics AppKinetics for communication.

Authentication Delegation
The feature for transferring authentication of the end user from one application to another. An application for which authentication is delegated does not display its unlock screen, and does not have its own security password. Authentication delegation can be used between two GD applications, and between GD applications and the GFE mobile client. Authentication delegation is controlled by the enterprise administrator through the management console of the respective software product, either GC or GFE Good Mobile Control.

C

CIFS
Common Internet File System - the standard way that computer users share files across corporate intranets and the Internet. An enhanced version of the Microsoft open, cross-platform Server Message Block (SMB) protocol, CIFS is a native file-sharing protocol in Windows.

CLI
Command Line Interface

COTS
Commercial Off the Shelf HTTP Proxy

D

DC
Direct Connect

DMZ
Demilitarized Zone

DMZ proxy for Direct Connect
HTTP proxy in the enterprise perimeter network that relays DC connections.

DN
For a single domain Active Directory Domain Service, this is the text box for the Distinguished Name (DN) of the starting point for directory server searches. For example: DC=mmycompany,DC=com. The Connector starts from this DN to create master lists from which you can later filter out individual users and groups. For a multidomain Active Directory Domain Service (AD DS) forest, the appropriate action is to leave this text box blank.

F

FQDN
fully qualified domain name

G

GC
Good Control server. The GD server component which hosts the web-enabled Good Control management console, or GC console, for managing permissions and settings for Good Dynamics applications. GC resides on a machine belonging to your organization.

GD

Good Dynamics. Good product that gives companies a set of development tools to create their own secure apps built on the technology used to create GFE.

GD Application ID

The unique identifier used throughout GD to identify the application for the purposes of entitlement, publishing and service provider registration.

GD Authentication Token mechanism

A token-based single sign-on feature that enables an end user to be authenticated by an application server without the need for entry of any further credentials.

GD Direct Connect

The feature for relaying GD communication through a proxy in the enterprise perimeter network (also known as DMZ or demilitarised zone) instead of through the GD NOC. This feature also enables GP servers to be deployed in the enterprise perimeter network, instead of behind the firewall.

GD Enterprise Servers

Two GD components installed behind the enterprise firewall: Good Control (GC) and Good Proxy (GP).

GD NOC

Good Dynamics Network Operations Centre - provides a secure communications infrastructure between the GD Runtime on the mobile device and the GD enterprise servers behind the firewall.

GD Runtime

The component that is embedded in a mobile application to enable its connection to the GD platform and container. Every GD application includes an instance of the Good Dynamics Runtime. Alternative form: Good Dynamics Runtime

GD SDK

Good Dynamics Software Development Kit. The products that enable developers to build GD applications from source code in the native programming languages of the mobile platform. Native source code includes, for example, Objective-C on iOS, and Java on Android. Other forms: Good Dynamics SDK Good Dynamics Software Development Kit

GD Shared Services

Framework for collaboration that includes Application-Based Services and Server- Based Services. Both types of service use a consumer-provider model. The consumer is always a GD application. The provider of an application-based service will also be a GD application. The provider of a server-based service will be an application server. Alternative forms: GD Shared Services Good

Dynamics Shared Services Framework GD Shared Services Framework Shared Services Framework

GD Wrapped Application

An application in which the GD Runtime has been embedded by using the GD Wrapping process. Other form: Good Dynamics Wrapped Application

GD Wrapping

The product for embedding the GD Runtime in a mobile application executable without requiring access to application source code. Other form: Good Dynamics Wrapping

GDN

Good Developer Networking. A web portal to support app development. • Download the Good Dynamics SDK • Download the Good Dynamics Servers • Access technical support, the Good Community, and other resources • Get notifications for technical updates • Get access to Good Dynamics enabled applications • Connect with developers and Good ISV partners

GEMS

Good Enterprise Mobility Server

GFE

Good for Enterprise

GNP

Good Notification Push. Protocol that allows notification messages to be pushed from an application server to GD app.

Good Dynamics AppKinetics™

Mechanism for secure exchange of application data between two mobile applications on the same mobile device. AppKinetics data exchange uses a consumer-provider model. One application in the exchange provides a service that is consumed by the other.

GP

Good Proxy. The GD server component which provides a secure bridge between the GC server and your enterprise application servers, if any exist, and delivers messages to and from GD applications. GP resides on a machine belonging to your organization.

GRP

Good Relay Protocol. Protocol for end-to-end secure communications between the GD app and the GP server.

GUID

Globally Unique Identifier - is a unique reference number used as an identifier and typically refers to various implementation of the universally unique identifier (UUID) standard. See UUID.

GW

Good Wrapping. The GD server component which can be used to wrap non-GD iOS applications with GD technology, allowing you to secure your applications without the need for additional programming or access to source code. GW resides on a machine belonging to your organization.

H

HTML/CSS/JS

Hypertext Markup Language, Cascading Style Sheet, and JavaScript, which are the languages used to code applications in the Adobe PhoneGap MEAP.

I

IDE

Integrated Development Environment

IOPS

Input/Output Operations Per Second (pronounced eye-ops) is a common performance measurement used to benchmark computer storage devices like hard disk drives (HDD), solid state drives (SSD), and storage area networks (SAN). As with any benchmark, IOPS numbers published by storage device manufacturers do not guarantee real-world application performance.

ISV

Indepdent Software Vendor - a third-party software developer or reseller who has executed a partnership agreement with Good.

J

JKS

Java keystore

JSON

JavaScript Object Notation, the format used for AppKinetics service definitions files. JSON is a standard.

K

KCD
Kerberos Constrained Delegation. A single sign-on feature that enables an end user to be authenticated by an application server that uses Kerberos, without the need for entry of further credentials.

KDC
Key Distribution Center. A logical component of the Kerberos infrastructure

L

LDAP
Lightweight Directory Access Protocol - a directory service protocol that runs on a layer above the TCP/IP stack

LUN
In computer storage, a logical unit number, or LUN, is a number used to identify a logical unit, which is a device addressed by the SCSI protocol or Storage Area Network protocols which encapsulate SCSI, such as Fibre Channel or iSCSI.

LUSE
Logical Unit Size Expansion

M

MAM
Mobile Application Management

MMC
Microsoft Management Console

MyTerm

O

OWA
Outlook Web Access

P

Provisioning ID

Part of the activation key that is the same for all GD applications activated by the same end user at the same enterprise. The provisioning ID is typically the end user's enterprise email address.

R

Relay Server

Server in the NOC that provides communications between the GD app and GP servers.

Repository

In GEMS-Docs, a repository is shared data source designated by a Display Name, a Storage Type (File Share or SharePoint), and a Path. Each repository is defined with user access permissions. Repositories can be further organized into Lists. When a repository is member of a list, it can inherent the user access permissions defined for the whole list.

RTT

Round trip time

S

SDK

Software Development Kit. Typically a set of software development tools that allows for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar platform.

Server Clustering

A feature within GD that enables enterprises to deploy groups of servers as single nodes in their GD infrastructure. The following servers can be deployed in clusters using this feature: GP, GC, application servers.

Server-Based Service

A GD shared service that is provided by application servers. A server-based service could use any communication technology, including HTTP or TCP sockets.

Service Discovery

Feature that enables a prospective consumer of a shared service to query for available providers of the service. The result of a service discovery query will be a list of GD applications, for an application-based service, or a list of servers, for a server- based service. Alternative forms: AppKinetics Service Discovery

Service provider registration

Activity of adding a GD application or application server to the list of providers of a particular service. The list of service providers is hosted in the GD NOC.

Share

In GEMS-Docs, a share is synonomous with a repository and can be one of two storage types: File Share or SharePoint. See Repository.

SPN

Service Principal Name

SSL

secure socket layer

T

TLS

transport layer security

U

UI

User Interface

UPN - User Principal Name

In Active Directory, this is the name of the system user in email address format

UUID

Universally Unique Identifier - an identifier standard used in software construction. A UUID is simply a 128-bit value. The meaning of each bit is defined by any of several variants. For human-readable display, many systems use a canonical format using hexadecimal text with inserted hyphen characters. For example: de305d54-75b4-431b-adb2-eb6b9e546014 The intent of UUIDs is to enable distributed systems to uniquely identify information without significant central coordination.

UX

User Experience